



TIMELINEZ

VOLUME 5 ISSUE 2

FEBRUARY 1987

\$1.00

THE JOINT NEWSLETTER OF THE THREE TIMEX-SINCLAIR
USER GROUPS IN THE SAN FRANCISCO BAY AREA
** EBZUG PUG SVSTUG **

LISTING ON YOUR PRINTER

by William J. Pedersen

Can you LLIST a single program line?
Can you LLIST without interference from standard graphics?
Can you print a listing of all lines you want to relocate so
you can create MERGE program sets?
Can you omit the LLISTing program itself from being LLISTed?
Does your interface do what you WANT it to do?
Can you FORMAT the LLISTing as you wish?

Why the &%#@ not??!

All these features represent deficiencies in commercial LLIST programs. Mostly it is caused by a lack of recognition that PRINT & LIST refer to the SCREEN but LPRINT & LLIST refer to a PRINTER. The two CANNOT be treated alike. The "advantage" of using tokens in print strings applies only to the 2040 Printer. No other printer is compatible with TS2068 tokens, nor is there any reason for them to be. Sticking with pure ASCII, with no extensions, lets you talk with almost ALL available printers.

The program LLISTed here has none of the problems listed. It converts common tokens to ASCII. It ignores color commands and number slugs which should not print anyway. It converts the few remaining unprintable characters to "?". ON, ERR, SOUND & RESET are treated as characters, not tokens. UDGs are not printable, they appear as "?".

You can LLIST any number of lines, in any order, from one to the entire BASIC. You can JUMP around to collect them.

The program is written in relocatable format. All relative jumps and the program location depend on a single KEYWORD (WK).

Move it anywhere you need for MERGE compatibility. Just be sure to update the value of Wk in the present LINE 7900. You might want to change the name from WK to something else.

Though the program is entirely in BASIC, it takes advantage of the powerful editing capabilities of the TS2068. It is incidental that you can follow the progress on the screen.

If you're frugal, and hate to waste paper, set Margin=0, Wide=the maximum (80?) for your printer, and set Wq=0 in LINE 7910.

If you're conventional, and want your listing to look as much like the screen as possible, set Margin=15, Wide=32, and Wq=0.

If you're really with it, and like your listings indented for maximum readability, try Margin=10, Wide=64, and Wq=1.

NOTE: THE ORIGINAL ARTICLE WAS ACCOMPANIED BY SIX EXAMPLES OF VARIOUS CONFIGURATIONS OF PRINTOUT LISTINGS. ONE EXAMPLE IS PRESENTED ON THE BACK PAGE.

NOTE:

**MULTI-TASKING ON THE TIMEX 2068,
USING Z80 INTERRUPT MODE 2**

BY FRED TEMPLETON (EBZUG)

Using interrupt mode 2 on the TIMEX 2068 is no easy matter. The normal use of interrupt mode 2 is accomplished by using a combination of the interrupt vector register providing the hi byte of an address and the interrupting device providing the lo byte of an address to be called. To utilize the 2068's 16ms interrupt for your own purposes you would normally have to use interrupt mode 1 in a machine language program that is on an EPROM in the cartridge dock chunk 0. You would then have some interrupt handling program at address 00056 to do the multi-tasking. It is also possible to use interrupt mode 1 from a RAM located program that turns on the EXROM and captures the interrupt from the bank switching process. This does limit the use of the BASIC system ROM without additional use of bank switching. There must be other techniques too.

One way to avoid all the bank switching problems is to use interrupt mode 2. To do this you must ignore the note #2 at the bottom of page 269 of the 2068's user manual. The I register may hold a value from 000 to 063 or from 128 to 255. For some reason the I register may not hold a value from 064 to 127, this may have something to do with the process of generating the video output. Before you can invoke interrupt mode 2 you must establish a 257 (not 256!!!) byte interrupt table in RAM somewhere starting at the beginning of a 256 byte page, ie. the first address must have a lo byte address of 000. Address 32768 is the first such address that you can use, as the hi byte address of this location is 128 which would be the number which would be loaded into the I register should you use this as the beginning of the interrupt table. A more likely place for the interrupt table would be nearer the top of RAM so that you could be using some BASIC program while the interrupt programs are running in the background. If you are using BASIC while interrupt mode 2 is turned on then you should protect the hi RAM by lowering RAMTOP to the next lower unused address. You must also make sure that the BASIC program does not POKE anything into the table thereby causing a crash.

Once you have decided where to put the 257 bytes you will put the same hi and lo address for all 257 of them, that will be the call address of the interrupt servicing routine. This limits the servicing routine to 190 locations in RAM where the hi and lo bytes of the address of the routine are the same. For example if the interrupt table contained 257 bytes with the value of 240 then the interrupt service program would have to be located at 61680 (240+256*240) this could be a JUMP to a more suitable address. The reason for this is that the system generated interrupt does not place a lo address on the bus as specified in normal interrupt mode 2 operation, so whatever address happens to be on the bus is used in conjunction with the I register to find the interrupt table. The normal restriction is that the address placed on the bus must be even-numbered, but you will not know the number on the bus so it could be odd. If it happens to be 255 then it is odd and the 256th and 257th bytes would be used for the call address which is why 257 bytes are required.

So now that you have an interrupt table properly addressed and an interrupt servicing program at some appropriate location there are some more things to do before you can get multi-tasking to work. One is to establish what I will call a vector table. It can be as long as you need allowing 2 bytes for each entry. If there are 200 (unlikely) routines you want to execute after each interrupt, then the table must be 400 bytes long. You will be POKEing (or LOADing from machine code) addresses into the table when you want a particular routine to be executed with each interrupt; then POKEing them back to 0 when you want them to stop. To turn a task on from BASIC, POKE the lo byte first and then the hi byte last. This is necessary incase an interrupt is received between the two POKEs. To turn it off you must POKE the hi byte to 0 first and then the lo byte last. If the combination of the servicing program and the called routines exceeds the 16ms interrupt period then it will be necessary to leave interrupts disabled. The BASIC program will only have the balance of the last interrupt period to run the BASIC interpreter, therefore if the interrupting sequence takes up 99% of the interrupt period then the BASIC program will run at only 1% of normal speed. If the interrupting sequence takes 101% of an interrupt period then the BASIC program will run 49.5% of normal speed, this threshold will cause several problems. If your BASIC program uses timing, then when ever a full interrupt period is exceeded the timing will be wrong unless you compensate for it. If your BASIC program is slowed to 1% normal speed it would probably be useless (unless you want to watch a BASIC program be single stepped).

The service program must then get the addresses from the vector table and CALL the routine if the address is non-zero. When the complete vector table has been checked then there is still one more thing to do. The last item of the servicing program should be a JUMP (not CALL) to address 00056 of the HOME ROM, this will complete the normal interrupt mode 1 routine which updates FRAMES, scans the keyboard, enables interrupts, and RETURNS to where it came from.

With everthing in place you still need a machine code routine to LOAD the I register with the hi byte of the first address of the interrupt table and to turn on interrupt mode 2. This must be the last thing you do. It is a good idea to SAVE everything first before you execute this. You might also want a routine to turn interrupt mode 1 back on, which turns interrupt mode 2 off without having to POKE the vector table back to 0's.

In summary when a interrupt is received the microprocessor PUSHes the next executable address onto the stack. The I register is used in conjunction with the bus to find the interrupt table. The interrupt table then points to the service program which in turn looks for addresses on the vector table to be CALLED. When all is CALLED then there is a JUMP to 00056 to complete the process and RETurn to the address on the top of the stack.

Why do all of this? I did it to be able to include a print spooler in the word-processor I wrote. It turned out so well I added an input buffer, clock/calendar, and clock display. Even with my word-processor located in chunk 1 of the Cartridge Dock, I can begin a print-out to my EPSON printer, print-out some other part of the text file to the 2040 printer, or QUIT the word-processor to the normal BASIC operating system and then LPRINT, LLIST, or COPY to the 2040 printer at the same time the EPSON is running. I have had up to five things going at one time in addition to the BASIC program and not noticed to much of a slow down of a BASIC program. To the right is a print-out of my assembly code to turn IM2 off, create the interrupt table, turn IM2 on, and the service program. Note that the vector table is located in the end of the nomally unused part of STRMS system variables. I use the first vector table address pair exclusively for the print spooler; the address is not CALLED if the EPSON printer is not ON LINE. I can also stop the print spooler with a three key break (CAPS/HIFT/SYMBL SHIFT/N).

For additional clarification call or write.
 Fred Templeton (415-754-5575)
 520 Northbrook Ct.
 Antioch, CA 94509

```

L6000 12600 89  INTERRUPT MODE 2
06000 ORG 23881;  TURN IM2 OFF
06100 IM 1
06200 RET
06250 /
06300 ORG 23884; INTERRUPT TABLE
06400 LD HL,64000;  H=250 L=0
06500 LD A,H
06600 LOOP LD (HL),93; AT 23901
06700 INC L
06800 JR NZ,LOOP
06900 INC H
07000 LD (HL),93;  257th BYTE
07100 LD I,A;  HI BYTE OF TABLE
07200 IM 2;  TURN IM2 ON
07300 RET
07400 / INTERRUPT SERVICE
07500 ORG 23901;  93+256*93
07600 PUSH HL;  SAVE REGISTERS
07700 PUSH DE
07800 PUSH BC
07900 PUSH AF
07910 EX AF,AF'
07920 PUSH AF
07940 / 3 KEY STOP PRINT-OUT
08000 LD A,254; BOTTOM LEFT ROW
08100 IN A,(254)
08200 RRA;  CAPS SHIFT KEY
08300 JR C,VECTORS;  NOT DOWN
08400 LD A,127;BOTTOM RIGHT ROW
08500 IN A,(254)
08600 AND 10;SYM SHIFT & N KEYS
08700 JR NZ,VECTORS;  NOT DOWN
08800 LD L,A=0
08900 LD H,A
09000 LD (23804),HL;STRM#15 OFF
09050 / PRINTER STATUS
09100 VECTORS LD BC,191
09200 IN D,(C);  EPSON?
09300 BIT 0,D;0=PRINTER ON LINE
09400 LD B,6; VECTOR TABLE SIZE
09500 LD HL,23605; HI STRMS#15
09600 JR Z,STRMS; PRINTER READY
09700 DEC B;  PRINTER NOT READY
09800 DEC HL;  IGNORE FIRST
09900 DEC HL;  VECTOR ENTRY
09950 / LOOK AT VECTOR TABLE
10000 STRMS LD A,(HL); HI BYTE
10100 OR A;  =0?
10200 LD D,A;  SAVE IT
10300 DEC HL;  GET LO BYTE
10400 LD E,(HL);  SAVE IT
10500 LD (call+1),DE;CALL ADDRS
10600 PUSH HL
10700 PUSH BC
10800 call CALL NZ,0;VECTORED!!
10900 POP BC
11000 POP HL
11100 DEC HL;  NEXT HI
11200 DJNZ STRMS;REPEAT B TIMES
11250 / TICK THE CLOCK
11300 LD B,1
11400 BIT 5,(IY+48);  FLAG52
11500 JR Z,CLK
11600 RES 5,(IY+48);  ADJUST FOR
11700 LD B,10;  DELETE REPEATS
11800 CLK PUSH BC
11900 CALL CLOCK;  AND CALENDAR
12000 POP BC
12100 DJNZ CLK
12105 /
12110 POP AF; RESTORE REGISTERS
12120 EX AF,AF'
12200 POP AF
12300 POP BC
12400 POP DE
12500 POP HL
12600 JP 56;  FRAMES, KSCAN, EI

```

Bob Orrfelt has brought it to our attention that some sections of the 2068 User Manual are inferior to the Spectrum Manual. Bob has supplied TIMELINEZ with a copy of the Spectrum Guide and from time to time we will reprint sections that may be of interest to our readers. This month we examine DEF FN.

You can also define functions of your own. Possible names for these are **FN** followed by a letter (if the result is a number) or **FN** followed by a letter followed by \$ (if the result is a string). These are much stricter about brackets: the argument must be enclosed in brackets.

You define a function by putting a **DEF** statement somewhere in the program. For instance, here is the definition of a function **FN s** whose result is the square of the argument:

10 DEF FN s(x)=x*x: REM the square of x

DEF is obtained in extended mode, using **SYMBOL SHIFT** and 1. When you type this, the computer will give you **FN** automatically, because in a **DEF** statement the **DEF** is always followed immediately by **FN**. After this, the **s** completes the name **FN s** of the function.

The **x** in brackets is a name by which you wish to refer to the argument of the function. You can use any single letter you like for this (or, if the argument is a string, a single letter followed by \$).

After the = sign comes the actual definition of the function. This can be any expression, and it can also refer to the argument using the name you've given it (in this case, **x**) as though it were an ordinary variable.

When you have entered this line, you can invoke the function just like one of the computer's own functions, by typing its name, **FN s**, followed by the argument. Remember that when you have defined a function yourself, the argument must be enclosed in brackets. Try it out a few times:

```
PRINT FN s(2)
PRINT FN s(3+4)
PRINT 1+INT FN s (LEN "chicken"/2+3)
```

Once you have put the corresponding **DEF** statement into the program, you can use your own functions in expressions just as freely as you can use the computer's.

Note: in some dialects of BASIC you must even enclose the argument of one of the computer's functions in brackets. This is not the case in ZX Spectrum BASIC. **INT** always rounds down. To round to the nearest integer, add 5 first - you could write your own function to do this.

20 DEF FN r(x)=INT (x+0.5): REM gives x rounded to the nearest integer.

You will then get, for instance,

```
FN r(2.8) = 3    FN r(2.4) = 2
FN r(-2.9) = -3  FN r(-2.4) = -2
```

Compare these with the answers you get when you use **INT** instead of **FN r**. Type in and run the following:

```
10 LET x=0: LET y=0: LET a=10
20 DEF FN p(x,y)=a+x*y
30 DEF FN q()=a+x*y
40 PRINT FN p(2,3),FN q()
```

There are a lot of subtle points in this program.

First, a function is not restricted to just one argument: it can have more, or even none at all - but you must still always keep the brackets.

Second, it doesn't matter whereabouts in the program you put the **DEF** statements. After the computer has executed line 10, it simply skips over lines 20 and 30 to get to line 40. They do, however, have to be somewhere in the program. They can't be in a command.

Third, **x** and **y** are both the names of variables in the program as a whole, and the names of arguments for the function **FN p**. **FN p** temporarily forgets about the variables called **x** and **y**, but since it has no argument called **a**, it still remembers the variable **a**. Thus when **FN p(2,3)** is being evaluated, **a** has the value 10 because it is the variable, **x** has the value 2 because it is the first argument, and **y** has the value 3 because it is the second argument. The result is then, $10+2*3=16$. When **FN q()** is being evaluated, on the other hand, there are no arguments, so **a**, **x** and **y** all still refer to the variables and have values 10, 0 and 0 respectively. The answer in this case is $10+0*0=10$.

Now change line 20 to

20 DEF FN p(x,y)=FN q()

This time, **FN p(2,3)** will have the value 10 because **FN q** will still go back to the variables **x** and **y** rather than using the arguments of **FN p**.

Some BASICs (not the ZX Spectrum BASIC) have functions called **LEFT\$**, **RIGHT\$**, **MID\$** and **TLS**.

LEFT\$ (a\$,n) gives the substring of **a\$** consisting of the first **n** characters.
RIGHT\$ (a\$,n) gives the substring of **a\$** consisting of the characters from **n**th on.
MID\$ (a\$, n1, n2) gives the substring of **a\$** consisting of **n2** characters starting at the **n1**th

TLS (a\$) gives the substring of **a\$** consisting of all its characters except the first. You can write some user-defined functions to do the same: e.g.

10 DEF FN t\$(a\$)=a\$(2 TO): REM TLS

20 DEF FN l\$(a\$,n)=a\$(TO n): REM LEFT\$

Check that these work with strings of length 0 or 1

Note that our **FN l\$** has two arguments, one a number and the other a string. A function can have up to 26 numeric arguments (why 26?) and at the same time up to 26 string arguments

Exercise

Use the function **FN s(x)=x*x** to test **SQR** you should find that

FN s(SQR x)=x

if you substitute any positive number for **x**, and

SQR FN s(x)=ABS x

whether **x** is positive or negative (Why the **ABS**?)

NEXT TIME : BEEP

The following letter was received from John Riley who is the S/W Librarian for the CATS User Group. John has given us permission to extend his offer for non-commercial S/W to all the TIMELINEZ readers. A listing of his programs can be found on the next page.

John Riley

1316 Farrara Dr.

Odenton, Maryland 21113

Thank you for showing your interest in the Capitol Area Timex-Sinclair User's Group program library. We are interested in expanding our stock of useful public domain programs by trading the ones we have for some that you have. Here is the deal we offer. If you will send us one or more useful and interesting programs on a tape in a Post Office-proof mailer and include return postage inside the mailer, I will put your program into our library and then record the volume of your choice from the three volumes listed below. I will also include our Spectrum catalog (small though it may be) if you will mention that you want it. By the way, good Spectrum programs are welcome, too! TAKE NOTE, HOWEVER: 1) The programs you submit must NOT be ones that are commercially produced. 2) You MUST include return postage inside your package, or I cannot get the programs back to you! 3) The programs MUST be for the 2068 or Spectrum! 4) The programs you submit MUST be of acceptable quality, not something that anyone could dash off in five minutes of programming!

Still interested? Great! Send me your goodies, tell me which volume of programs you want, and together we will increase each other's enjoyment and usefulness of our 2068 machines!

Sincerely Yours,


John Riley

###Volume One --- Games

DATE -- A sham computerized dating service program that tends to match people up with the likes of Attila the Hun as their most perfect date. Great fun at Valentine's Day Parties!

SAILING -- An interesting little simulation game by our own Mark Fisher.

TRAIN -- Novel use of sound and graphics in this little program. Good train wreck!

KENDO -- A computer adaptation, we are told, of an ancient Chinese betting game.

SPACEWAR -- Another Mark Fisher creation. We are invited to blow the opposition to dust with proximity bombs. Not just a shoot-em-up, though.

TAG -- A frustrating game in which we are to get a reluctant colored square to go where we want it.

EGG SAVIOR -- Arcade style. Rescue the eggs from nasty laser guns.

HUNCHY -- Arcade in which you are the Hunchback of Notre Dame. Rescue Esmeralda!

STAR TREK -- The CLASSIC game that everyone used to sneak onto the mainframes at college in the days of yore.

BRICK-YARD BILL -- Arcade-style, help Bill stack bricks without hemming himself in. Quite difficult.

BRIDGE -- Build a bridge by dropping sections from a helicopter before a blind man can fall into the river.

TWISTS -- Restore a 4x4 Rubic's Cube face to its original configuration before the time runs out. Very challenging.

MINOTAUR -- Find the exit to a maze, collecting invisible objects along the way, before the Minotaur gets you!

DEATHRACE -- Run down innocent pedestrians with your race car. Cute little tombstones mark the scene of their demise.

PEAKOUT -- A Basic version of the classic Atari game.

ANGELS -- A space-invaders arcade game.

MOONLANDER -- Land your lander safely on the Moon. I went "SPLAT" every time.

###Volume 2 --- Educational & Utilities

SPANISH TUTOR -- Self-explanatory.

GEOGRAPHY -- An unfinished states and capitals program. Someone fill it out, please!

COLOR PLOT -- A drawing program aimed at children's use.

COLOR DOODLE -- A variation on the above program that makes blockier pictures.

DRAW -- A more sophisticated drawing program.

SPECDRAW -- An even more sophisticated graphics program, featuring 18 sizes of text, fill, four-point banding and more. You really need a keyboard overlay for this one!

PIXPIX -- The winner of the Syncware News contest for graphics software. Gives you pixel by pixel control of the screen.

WOLVERINE -- A sample picture drawn on PixPix.

RANDOM FAN -- Pretty!

PIE CHART -- Makes pie charts from data you supply.

3D SHAPE -- Draws 'em.

BARCHART -- Generates a barchart from your data.

HEADLINE and BANNER -- Both of these generate large-size words running side-ways on a 2040 printer.

EDITOR -- A utility for creating user-defined graphics.

G -- Does the same as the program above, with a slightly different approach.

CHARACTERS -- Throws the 2068 character set onto an 8x8 grid, and gives the numerical value of each line of pixels.

64 COLUMN -- A basic program that shifts the 2068 into 64-column mode. Too slow to be useful, but it is educational.

Volume 3 --- More Utilities

HEADSCAN -- A header-reader program that is a must for anyone who wants to transfer multi-load programs to another tape or disk format.

SCANNER -- A very interesting program that takes an audio input from the MIC jack and gives a running graphic reading of it.

VIDEofile -- A computerized filing system for your videotape library.

TAPECHED -- Computerizes your weekly videotaping schedule.

BANKREC -- a computer checkbook balancing program.

LISTS -- Simply the best public domain data base that I know of for the 2068. The only way to beat it is to buy ProFile.

FORMULARY -- A darkroom-chemistry mixing guide for photographers who do their own developing.

FIRSTLOAD -- the mc program that translates TS1000 programs into 2068 programs.

MCEDIT -- Not McDonald's latest offering, but rather a simple machine code editor.

MCLOADER,-- A simple mc loader.

BOLD -- Redefines the 2068 character set into fatter letters and symbols. A 2040 printer will then print them that way.

FLIP -- Rotates a screen 90 degrees every time you press the button!

CRYPTO -- Creates a simple substitution cypher that you define.

HEX -- a hexadecimal loader.

HEXSPEC -- a Spectrum hex loader. I s'pose it would work with a 2068 too.

TOMORROW -- a primitive word processor that has a nice personality.

INTERPOLATE -- interpolates a graph from the data you supply.

***** NOTE ***** THE NEXT THREE PROGRAMS ARE SOME OF JACK DOHANEY'S FAIRWARE. IT IS GOOD STUFF!! I URGE YOU IF YOU FIND YOURSELF USING THESE TO SEND A FEW DOLLARS TO JACK IN GRATITUDE FOR DOING THESE THINGS FOR US.

EXTENDED BASIC -- Adds nine new commands to our standard basic, which significantly enhance the 2068's sound and animation facilities.

KEYWORD -- Disables the single-stroke keyword system and allows one to type in Basic commands directly from the keyboard, just like the Other Guys' computers.

TOOLKIT -- a programmer's utility which includes Renumber, Remkill, and a Compactor.

THE SPECTRUM CATALOG !! (SHORT & SWEET)

ANIMATE -- a graphics utility which generates a 3D picture from your 1D bisection, rotates and recreates this picture 8 times, and then pages these pictures sequentially to achieve a spinning animation effect.

GALACTOIDS -- A space-invader type of game that is fast, furious, and infuriatingly difficult.

BLOCKMAN -- A more sedate but still challenging arcade game that features horizontal scrolling of subsets of the game board.

PUG NEWS
by Norm Lehfeldt

**ARRIVAL OF SCLOD CHIPS FOR 2068
HIGHLIGHTS JANUARY PUG MEETING**

The long-awaited SCLOD chips for the 2068 were distributed by Pat Morrissey to those who had ordered them at the November meeting. These are the custom chips that are the heart of the 2068 and which cannot be obtained from usual electronic parts sources. Pat is retaining two of the chips for club use and reports that some are still available from CATS for those who may wish to have a little hardware insurance against the time when Little Rock stops doing repairs.

Also at the January meeting:

Rita Toth reports that the prison project is back on track after some setbacks due to troubles at the prison. Equipment we have supplied is getting good use.

Rita has now been handling this project for a year now and it is time to pass it along to someone else. This has been a very successful project and a volunteer is needed to continue it.

Plans for the May T/S Computer Fest in Indianapolis are shaping up. The organizers are soliciting contributions and other support from groups throughout the country. It is to our advantage to participate in any way that we can, since the success of such events will prolong the life of our favorite computer. Paul Holmgren of ISTUG is the Executive Director of the Fest. We will be sending sets of back issues of 'TIMELINEZ' to the event.

Bob Orrfelt is working on an enhanced ZX81 based on the kits available at Sunset Electronics. It will have a 16K EPROM in place of the original ROM and 32K internal memory.

Treasurer Pat Morrissey reminds everyone to pay dues. Each issue of 'TIMELINEZ' costs us about \$65.00 to print and mail.

Bill Miller is urging Silicon Valley Group members to attend PUG meetings. There are some problems with meeting sites for the Silicon Valley Group. Bill also says that a T/S Special Interest Group is being established as part of the Computer Workshop at Stanford. News about the workshop is available on the Public Domain Software Exchange -- 408 735 7390 -- 300/1200 baud.

Walt Johnson demonstrated his 4" Sony Watchman TV/Monitor as a screen for the 2068. It produces readable displays with MSCRIPT -- and it matches the tasteful Timex silver livery!

Next PUG meeting on February 15.

SVSTUG NEWS
by Bill Miller

1. Upcoming Meetings:

- 1.1 January 27, Tuesday, 7:00 P.M.; El Camino Hospital, Meeting Room A (in basement), 2500 Grant Road, Mountain View.
- 1.2 February 15, Sunday, 1:00 P.M.; Peninsula Hospital, Trousdale Meeting Room (in basement), Burlingame.
- 1.3 March 15, Sunday, 1:00 P.M.; Peninsula Hospital.
- 1.4 Every Saturday, 9:00 A.M.; Stanford University, Jordan Hall (in the Quadrangle), Math Building (Number 380) in room 380C (downstairs). For topic, check the PDSE (Public Domain Software Exchange) BBS in Sunnyvale at 408 735-7390, 300/1200, 8, N, 1, 24 HRS, 32 lines (no waiting!).

2. GUTS/SV Meeting News:

- 2.1 The 861230 GUTS/SV (SVSTUG?) meeting was lightly attended. Maybe it was because of the new location or the holidays. Mark Wahl reported he has completed the design of the message handler for his TS1000 BBS. It will use variable length records to store the messages in the 32K-64K memory locations of the TS1000. Mark intends to pattern his modem interface after the one used in Randy and Lucy Gordon's 'Tiny Board' BBS. If you want current information on how Mark is progressing you might try leaving him a message on the Sunnyvale PDSE BBS.

- 2.2 The 870110 GUTS/SV (TCW TS SIG (The Computer Workshop Timex Sinclair Special Interest Group)?) featured a demonstration of the latest hardware and software for the Amiga, given by Claus Wiebe.

Claus demonstrated a 2MB expansion unit for the Amiga and also the Amiga's 4096 color capability.

Bob Orrfelt reported he is working on a TS1000/ model railroad controller unit. This unit will utilize the 32K x 8 Static Ram memory chips (\$19 at FRY'S) with battery backup.

Bob also provided SLIX with information on the User Friendly Research boards: ZXTALKER, ZXPANDER, REAL TIME CLOCK, & A/D CONVERTER.

These were donated to the TIMELINEZ groups about 851202 by Leland Harker?

3. SLIX (Sinclair Information eXchange) needs help! We have been using 'Master Tool Kit' from OCP to help with the Newsletter Exchange, but it uses up 9K and we don't need all it's features.

Also, if we load 'Master Tool Kit' from WafaDrive we can't get it to operate, so we have to load it from cassette. Yuk! What we need, is the features we have found most useful from 'Master Tool Kit' plus a couple of additional ones as either BASIC or Machine Code routines or BASIC extensions such as the WafaDrive, ZX Microdrive, or Timex Portugal disc add. Features needed are:

- 3.1 Delete- Since we are operating in the Spectrum mode we can't use the TS2068 DELETE.
- 3.2 Renumber- with range, starting line number, & increment able to be specified.
- 3.3 Find- but unlike 'Tool Kit', the result can be assigned to a BASIC variable- either line number or line contents.
- 3.4 Sort- We keep our mail list as DATA statements. Each record (Exchange Newsletter address and info) is stored as one DATA statement. It would be nice to be able to sort them.
- 3.5 Write- The opposite of READ, take a BASIC variable (string) and put it in a DATA statement.

If you would like to help, contact: Bill Miller,
408 253-3175, 6675 Clifford Dr., Cupertino, CA,
95014-4530.

EBZUG NEWS

BY FRED TEMPLETON

REPORT ON MEETING OF JANUARY 22, 1987

The meeting was attended by approximately 15 members, most of which arrived in dry weather. Those who stayed late got soaked by the rain when they left.

After lengthy discussion it was decided by vote to include "TIMEX-SINCLAIR" in the official name of our group. It was felt that the more informative name would help potential members find us. USER GROUP listings in COMPUTER currents, THE MICROTIMES, and COMPUTER SHOPPER, etc which identify us only by initials may not be inviting for those who are looking for us.

The next meeting date of February 26th was confirmed with the library office. Future meetings at the library seem to be questionable. At least three of our members have contacted the librarian and have had no luck in booking the room that we now call home. Unless the situation changes with the library it was concluded that we would invite ourselves to have a joint meeting with PUG at their regularly scheduled meeting, which is expected to be Sunday March 15th at 1:00 p.m. Many of us would like to meet the members of the other TIMEX user groups, and it was suggested that we have joint meetings at least a couple times a year. In the mean time every one of us must now be on the look-out for a new home: an office that we could use, a saving and loan conference room, or almost any place with four walls, lights and 120V outlets. If everyone does some serious searching we will surely find a new place, even if only for a few meetings until we can re-establish ourselves with the library.

***** FEBRUARY MEETINGS *****

15 PENINSULA USER GROUP 1:00

Peninsula Hospital
1783 El Camino Real
Burlingame

15 SILICON VALLEY USER GROUP 1:00

Peninsula Hospital
1783 El Camino Real
Burlingame

26 EAST BAY USER GROUP 7:30

West Branch Library
1125 University Avenue
Berkeley

***** MARCH MEETING *****

15 JOINT MEETING

ALL USER GROUPS
Peninsula Hospital
1783 El Camino Real
Burlingame 1:00

EBZUG EAST BAY Z80 USER GROUP
===== 3128 KING STREET
BERKELEY, CALIFORNIA 94703
(Woody McPheeter's)

CONTACT: RUSS ENGLISH (415) 465-3116

MEETINGS: FOURTH THURSDAY OF EACH MONTH, 7:30 P.M.
WEST BRANCH LIBRARY
1125 UNIVERSITY AVENUE, BERKELEY

MAKE CHECK FOR DUES PAYABLE TO "WOODY MCPHEETERS".

PUG PENINSULA USER GROUP
===== 311 MICHELLE LANE
DALY CITY, CALIFORNIA 94015
(415) 878-1773

PRESIDENT: GEORGE MCKRIDGE

MEETINGS: THIRD SUNDAY OF EACH MONTH, 1:00 P.M.
PENINSULA HOSPITAL
1783 EL CAMINO REAL, BURLINGAME

MAIL DUES TO "PAT MORRISSEY", 2000 CRYSTAL SPRINGS ROAD,
BLDG 21, APT 22, SAN BRUNO, CA 94066

SVSTUG SILICON VALLEY SINCLAIR TECHNOLOGY USER GROUP
===== 6675 CLIFFORD DRIVE
CUPERTINO, CALIFORNIA 95014
(408) 253-3175

PRESIDENT: RITA CARR, (408) 738-2888, X-4579

MEETINGS: LAST TUESDAY OF EACH MONTH, 7:00 P.M.
(FOR LOCATION OF MEETINGS, SEE ACCOMPANYING
CALENDAR AND SVSTUG NEWS.)

MAKE CHECK FOR DUES PAYABLE TO "SINLINK".

THIS TIMELINEZ NEWSLETTER IS A JOINT PUBLICATION OF THE THREE
TIMEX-SINCLAIR USER GROUPS IN THE SAN FRANCISCO BAY AREA.

NEW MEMBERS AND VISITORS ARE ALWAYS WELCOME.

FOR FULL MEMBERSHIP (WHICH INCLUDES PARTICIPATION IN GROUP
MEETINGS, THE TIMELINEZ NEWSLETTER, TAPE LIBRARY PRIVILEGES,
AND SPECIAL EVENTS), SEND \$15 ANNUAL DUES TO ONE OF THE ABOVE
ADDRESSES WITH THE CHECK MADE PAYABLE AS INDICATED.

FOR JUST AN ANNUAL SUBSCRIPTION TO THE NEWSLETTER, SEND A \$10
CHECK TO ONE OF THE ABOVE GROUPS.

WLLIST PROGRAM LISTING

ENTER (AS TOKENS): ON ERR RESET

```

7900 LET WK=7900: LET Margin=10: LET Wide=64: LET W1n=2: C GO TO
WK+19: DIM A$(Margin)
7901 C7: INPUT PAPER 5;" LLIST "; PAPER 7;"From LINE: ";WFirst;""
to ";W1st
7902 LET Wa=PEEK 23635+256*PEEK 23636
7903 LET Wn=256*PEEK Wa+PEEK (Wa+1): LET W1=PEEK (Wa+2)+256*PEEK
(Wa+3): IF Wn<WFIRST THEN LET Wa=Wa+W1+4: GO TO WK+3
7904 IF Wn>W1st THEN GO TO WK+17
7905 CLS : PRINT TAB 4-LEN STR$ Wn;Wn;" ";: FOR n=Wa+4 TO Wa+W1+
2: LET W=PEEK (n): IF W=14 THEN LET n=n+5: GO TO WK+9
7906 IF W>15 AND W<22 THEN LET n=n+1: GO TO WK+9
7907 IF W>126 AND W<165 THEN LET W=63
7908 PRINT CHR$ W:
7909 NEXT n: LET Wz=800-PEEK 23688-32*PEEK 23689: PRINT PAPER 5;
" ";Wz+1;
7910 LET W9=0: GO SUB WK+15: FOR n=0 TO Wz: LET J=INT (n/32): LE
T K=n-32*XJ: LET L$=SCREEN$ (J,K): IF L$=":" AND W9>0 THEN LPRINT
CHR$ 13: GO SUB WK+15: LPRINT " ";: LET Wc=5
7911 LPRINT L$;: LET Wc=Wc+1: IF L$="**" THEN LET W9=-W9
7912 IF Wc>Wide THEN LPRINT CHR$ 13: GO SUB WK+15
7913 NEXT n: LPRINT
7914 LET Wa=Wa+W1+4: GO TO WK+3
7915 LPRINT A$;: LET W1n=W1n+1: LET Wc=1: IF W1n>58 THEN LET W1n
=0: LPRINT CHR$ 12: GO TO WK+15
7916 RETURN
7917 INPUT "Another section? Yes or No: ";L$: IF LEN L$ THEN IF
L$(1)="Y" OR L$(1)="y" THEN GO TO WK+1
7918 STOP : GO TO WK
7919 LET A$="": GO TO WK+1

```

ENTER (AS A TOKEN): ON EP

MANAGING EDITOR: Walt Gaby

EDITORS: EBZUG John Ezike
PUG G. Mockridge
SVSTUG Bill MillerADVERTISING: Norm Lehfeldt
Bill Strick

N/L EXCHANGE: G. Mockridge

ADVERTISING SCHEDULE:

Full Page	\$40
Half Page	\$25
1/4 Page	\$15
1/8 Page	\$10

TIMELINEZ
P.O. BOX 1312
PACIFICA, CA 94044
USA

First Class Mail